AFRL-IF-RS-TR-2006-354
Final Technical Report
December 2006

INFOSPACE CONCEPT EXPLORATION AND DEVELOPMENT ACROSS SECURE COMMUNITY OF INTEREST (COI) BOUNDARIES

Sensis

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Rome Research Site Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-IF-RS-TR-2006-354 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/

DALE W. RICHARDS
Work Unit Manager

/s/

JAMES W. CUSACK
Chief, Information Systems Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
DEC 2006	Final	Sep 05 – Sep 06

4. TITLE AND SUBTITLE

INFOSPACE CONCEPT EXPLORATION AND DEVELOPMENT ACROSS SECURE COMMUNITY OF INTEREST (COI) BOUNDARIES

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA8750-05-C-0269

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Nick Kowalchuk

5d. PROJECT NUMBER
ICED

5e. TASK NUMBER
05

5f. WORK UNIT NUMBER
04

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Sensis
85 Collamer Crossings
East Syracuse, NY 13057

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFSE
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2006-354

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 06-801

13. SUPPLEMENTARY NOTES

14. ABSTRACT This effort developed technology to securely compose, maintain and dissolve infospheres, including aspects of service discovery, negotiation, configuration, revocation, and policy enforcement, and basic actions associated with composing and decomposing COIs. A capability was developed to automate the announcement, discovery, request matching, and life-cycle management of information services both within and across security domain boundaries. Requirements and design trade-offs associated with the management of information services across security domains were researched and documented. A requirements analysis indicated that in a multi-domain environment in which information services must be shared between domains, the security framework should include location transparency of both service registries and service providers between the domains. An architecture was defined that enforces location transparency while enabling secure sharing of information services via new methods for secure announcement across security domain boundaries in a manner that allows only intended recipient domains to decrypt the service announcement. The major contribution of this work was the development of a new Trust Model for Secure Service Management incorporating secure service announcements, secure Service Manager processes, a secure Private Registry and an open Application Programming Interface (API) for secure Service Invocation requests. The resulting technology was demonstrated for a combined Federal Aviation Administration (FAA) and DoD scenario regarding Special Use Airspaces.

15. SUBJECT TERMS

OIM, COI, Community of Interest, Security, Infosphere, Architecture, Service

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dale W. Richards
U	U	U	UL	56	19b. TELEPHONE NUMBER (*Include area code*)

Table of Contents

List of Figures

List of Tables

1 Executive Summary

This Final Technical Report documents exploratory research and development focused upon the definition of currently unavailable technology to enable highly secure announcement, discovery, negotiation, and life cycle management of information services, both within a single security domain, and across multiple security domains.

The research problem was focused specifically on the identification of requirements that enable the information services required by all participants in a Community of Interest to be dynamically managed (announced, discovered, negotiated, monitored, and released) across security domain boundaries.

The research data documented in this report gives evidence that current approaches do not adequately address all key security considerations of this problem, leaving vulnerabilities that when exploited by attackers, leads to potential compromise of internal system hosts and theft of data. The research data indicates that current approaches (and architectures) have addressed overall information security in a manner that still leaves significant vulnerabilities in system security. Trust in current architectures is more decentralized, but still open to exploitation of:

- Location of service providers.
- Location of service registries.

Misuse of this "free" information can lead to direct attacks (both internal and external) that exploit vulnerabilities in the underlying network and host operating systems of any participating security domain.

The requirements analysis indicated that in a multi-domain environment in which Information Services must be shared both within and across security domain boundaries, a security framework should include location transparency of both service registries and service providers across security domain boundaries. A key contribution of this effort was the definition of a comprehensive framework that includes all of these security considerations, to eliminate the vulnerabilities of current approaches.

Based upon the results of the requirements analysis, a system architecture was defined for a Secure Service Management (SSM) system, including an open interface to an externally available Service Invocation System that is responsible for invoking information services. In addition to providing location transparency of both service providers and service registries, the recommended architecture included a key technical contribution detailing a new method for securely sharing information service announcements securely across security domain boundaries in a manner that allows only intended recipient domains to decrypt the service announcement. The system architecture was decomposed into a functional architecture for the Secure Service Management system, providing a detailed concept of execution among the system functions and a definition of function responsibilities.

A proof-of-concept demonstrator was developed to implement the design in software. To validate the demonstration system, an operationally realistic use-case was defined. The use case reflects a near-term operational need that exists between Federal Aviation Administration (FAA) and Department of Defense (DoD) Air Traffic Control (ATC) authorities to dynamically monitor the establishment of military Special Use Airspace (SUA) ranges established for training and exercises, and adapt FAA flight corridors such that commercial aircraft avoid active SUA ranges. The use case demonstrates a collaborative, yet restricted, sharing of critical information services across security domain boundaries that would be required to provide a secure SUA management and dissemination capability.

A final demonstration of the proof-of-concept system was performed at AFRL on 29 September, 2006.

Overall, this effort:

- Identified key gaps in the state of the art with respect to secure sharing of information services both within and across security domain boundaries.

- Defined an architecture that enforced location transparency while enabling secure sharing of information services via new methods for secure announcement and discovery of services.

- Developed a software proof-of-concept prototype implementation of the design.

- Successfully demonstrated feasibility of the concept using the prototype to validate an operationally-realistic use case.

Given the results of this effort, the recommended course of action for decision makers includes:

- Pursuit of Advanced Technology Demonstrations to further demonstrate operational utility. The proof-of-concept prototype system developed in this effort is extensible to participation in field exercises and advanced technology demonstrations that require secure sharing of information services between multiple domains.

- Advanced development of the service matching algorithm. The service matching algorithm provides a richer discovery interface for identifying all available services that may be relevant to a user's needs. The initial matching algorithm may be improved to provide a more granular hierarchical result set. This represents a contribution to advancing the state-of-the-art in discovery of services.

- Automation of the service requestor interface to include a machine-to-machine use case. The current implementation provides a human-to-machine interface for discovery and selection of available information services that a user requires. This interface can be adapted for the machine-to-machine case, where software applications may directly interact with the service request interface to discover, select, and bind to available services that meet a set of prioritized criteria for the service attributes required.

2 Introduction

This report is relevant to all DoD and commercial entities seeking technical solutions for the highly secure sharing of information services, both within and across security domain boundaries.

This section of the report introduces the subject, purpose, and scope for the effort. An outline of the remainder of the report is provided at the end of Section 2.

2.1 Subject

The subject of this effort is the automated sharing of information services between security domains, which includes all of the following functions:

- Secure announcement of available information services within a security domain, and a restricted set of shared services between a local domain and external domains.
- Discovery of all information services available to a requestor, regardless of which domain the service providers reside in.
- Service request matching, which determines which available services best fit the desired attributes of a service request, and ranks the matching results.
- Definition of a standard interface to an external service invocation system (which grants/denies access to information services and instantiates service data flows).
- Stateful monitoring of invoked services.
- Release, or termination of services upon successful fulfillment of terms or upon violation of terms.

2.2 Purpose

The purpose of this effort is to define, prototype, and validate a recommended architecture for cross-domain sharing of information services that closes gaps which introduce security vulnerabilities in current state-of-the-art.

2.3 Scope

The scope of this effort includes the following:

- *Research* current state-of-the-art approaches and standard methods for the secure sharing of information services.

- *Analyze* security vulnerabilities in current approaches, identify gaps, and recommend a technical architecture that closes the identified vulnerability gaps.

- *Design* the recommended architecture by decomposing the requirements into a functional specification for the software necessary to implement them.

- *Prototype* the recommended system by generating a detailed software design and developing the software.

- *Demonstrate* the prototype system per an agreed-upon demonstration scenario script that captures an operationally-realistic use case.

2.4 Report Structure

The remainder of this report is structured as follows:

Section 3 documents the research problem, requirements analysis effort, recommended system design, and proof-of-concept demonstration.

Section 4 describes the results and a discussion of their significance.

Section 5 interprets the findings and presents preliminary conclusions.

Section 6 documents the recommendations and course of action based on the results.

Section 7 is a list of Terms used in this document and their definitions.

Section 8 is a list of Symbols, Abbreviations, and Acronyms used in this document.

3 Methods, Assumptions, and Procedures

This section of the report comprehensively documents the work performed on the project. Subsection 3.1 describes the research problem, the requirements analysis performed, and the architecture that was defined to meet the requirements. 3.2 presents the system interface requirements that were identified. The system functional and physical architectures are described in 3.3 and 3.4, respectively. 3.5 identifies the software requirements, and 3.6 references the software detailed design. The proof-of-concept demonstration is discussed in 3.7.

3.1 Research, Requirements Analysis, and Architecture Definition

This section provides a comprehensive summary of the research portion of this project.

3.1.1 The Research Problem

The research portion of this effort focused specifically on the identification of requirements that enable the information services required by all participants in a Community of Interest to be dynamically managed (announced, discovered, negotiated, monitored, and released) across security domain boundaries.

This subsection of the report:

- Defines the context of the research performed.
- Reviews the state-of-the-art methods and specifications for addressing security of information services.
- Identifies a core set of requirements for security of information services.
- Identifies gaps in current technology with respect to meeting the core security requirements.
- Derives a new model for automated secure management of information services that addresses the stated vulnerabilities.

Overall, this subsection makes the assertion and gives evidence that current approaches do not adequately address all key security considerations of this problem, leaving vulnerabilities that when exploited by attackers, leads to potential compromise of internal system hosts and theft of data. A new security model is defined to address the technology gap.

3.1.1.1 Relevant Context

It is important to clarify an understanding of the context of the research, and the terms that shall be used to describe its core focus.

3.1.1.1.1 Information Services

An information service is an "offering" of structured data from a producer to eligible consumers. The service is described using a structured specification for what is "offered" by the producer, which we call a Service Specification. A Service Specification is comprised of all necessary attributes that define the characteristics of the service offered. Service Specifications must be meaningful to the potential consumers of a service, such that consumers may unambiguously define their information needs in a request.

In this effort the "producer" of an information service may be either:

- Archival (e.g. – a filesystem, or a database)
- Active (e.g. – a software process continually or periodically producing data)

In either case, the information Service Specifications for archival and active service providers is service type-dependent and defined individually according to the types of information offered. The choice of service attributes are a matter of consensus within a particular Community of Interest.

The design for integration of external information brokers (such as the Joint Battlespace Infosphere (JBI) broker model, IBM Websphere, etc.) within an information services management framework is out of the scope of this effort. Each information broker architecture has its own Application Programming Interface (API) for information requests, most are not cognizant of Security Domains (as defined in 3.1.1.1.3), and most do not provide methods for generating information Service Announcements describing the types of dynamic content that they broker. Each broker architecture would require a customized design approach to address the way that it would publish information Service Specifications and receive Requests for service invocation. In this effort we provide a recommended open API for publishing information Service Specifications, and for receiving Requests for service invocation, that may be utilized by external information brokers.

In heterogeneous domains, differences may exist in Service Specifications, regarding each domain's choice of how to specify service type and service attributes. This effort does not address the ontological approach to mapping between such differences. For the purposes of this effort, it is assumed that a COI (which shares a common business process model and associated vocabulary) defines a common ontology and adopts an agreed-upon set of Service Specifications for the information shared between domains. Alternatively, much existing research has already addressed ontology-based mapping between specifications, and may be used to handle the mapping issue across domain boundaries. In this effort, each domain will apply a consistent Service Specification to its service providers of the same type.

Clients

Clients are software applications or processes that operate on behalf of the users participating in a COI. Clients may be producers or consumers of information, or both. Throughout the remainder of this document, clients are synonymous with the term "Requestor".

3.1.1.1.2 Security Domain

A Security Domain is a core principle in the context of this project. A Security Domain is comprised of all users, hardware, and software processes that operate under the control of a single security policy, and are physically reachable on an IP-based subnetwork. The Domain's security policy is comprised of all the individual policy rules that govern the flow of information within the local security domain, and between the local domain and external trusted domains. All hosts in a security domain must be reachable (addressable) within the domain. Note that an organizational entity may be comprised of several Security Domains.

A "trusted external domain" is any Security Domain outside of the local Domain that has some mutually agreed-upon need for information exchange with the local Domain, and a set of associated security policy rules governing the limits of that information exchange between the trusted Domains. No access is allowed between trusted Domains other than that which is explicitly granted by corresponding rules in each participating Domain pair.

3.1.1.1.3 Stages of Service Management

Irrespective of any security considerations (which will be discussed later in the context of the approach), the process of exchanging services between producers and consumers is functionally comprised of the following stages:

3.1.1.1.3.1 Information Service Specification

An information Service Specification (introduced in 3.1.1.1.1) must exist for all services to be exchanged.

Ontology, semantics, and syntax as they apply to information Service Specifications is an interoperability issue that exists among service providers and consumers in heterogeneous systems. It is out of the scope of this effort to address a solution to the global ontology-mapping problem between domains that exchange services. In this effort, it is assumed that within a COI a mutually agreed-upon ontology and one corresponding service specification exists for services of each type.

The content of an information Service Specification is comprised of Service Attributes. The Service Attributes are metadata labels that describe all relevant characteristics of the service being offered. Each Service Specification contains a set of core attributes (that must be specified), and additional optional attributes that may apply to the service type. Within a Service Specification, the individual core Service Attributes and optional attributes may each be defined to have one acceptable value, or may be defined as a range of acceptable values.

This effort is focused on definition of a means to securely maintain Service Specifications in a local Service Registry, and to securely share Service Specifications across Security Domain boundaries during the Service Announcement process.

3.1.1.1.3.2 Service Announcement

Service Announcement is concerned with how a service provider makes its information Service Specification known to the entities that are intended to access it. A Service Registry is a repository for maintaining information Service Specifications. Service providers publish their information service specifications to Service Registries, thus "announcing" their service to consumers.

This effort is focused on defining a means to generate and securely distribute information Service Announcements both within and across Security Domains.

3.1.1.1.3.3 Service Discovery

Service Discovery is concerned with how service consumers find the information services that they can potentially access and bind to. Beyond that, Service Discovery is more specifically concerned with finding the information services that match a set of Service Attributes that a consumer is interested in to satisfy their information needs.

Service Discovery may be performed in the following ways by a Requestor:

1. A Requestor's application may interact through an interface that presents a structured representation of the contents of a Service Registry to enable selection of candidate services.

2. Via a Graphical User Interface, a Requestor may view the information services residing in a Service Registry and filter according to attributes that they select. This effort will focus on a means to securely access the contents of a Domain's Service Registry, and allow a Requestor to view and request available information services, without divulging sensitive location information across Security Domain boundaries.

3.1.1.1.3.4 Service Matching

Service requests may be issued by Requestors to the APIs of specific service providers who advertise their information services in a Service Registry. Prior to a consumer application binding to an information service, that consumer (or their application) must first decide which service or services advertised in a registry best meet their information needs.

Service Matching is a process through which a consumer's request for an information service is mapped (i.e. - "matched") to the specific provider or providers that can best satisfy all the attributes of the requested service. For a given Service Type, the service request processing system must map all service request attributes against the advertised attributes of active services, and determine a valid result set that best fulfills the request. Service matching is concerned with

the heuristics of how to generate a set of available services (from the set of all available services) that best fit a given service request. The matching process for a given request is based upon the service type, and the desired attribute values (or range of values) within that type that are selected by a Requestor as part of their service request.

The whole premise of Service Matching is based on the fact that:

- There may be multiple service providers that satisfy the attributes of a request.
- There may be some service providers that partially match the attributes of a request.

In either of the above cases, Service Matching has the responsibility to choose the options that best satisfy each request.

In the context of this effort, Service Matching provides a set of matching results (as a list of available services that most closely fit the service request). From the result set, the Requestor selects the service or set of services that they choose to accept as a form of service agreement. This agreement essentially takes the form of a contract between the service consumer and the service provider.

This effort will focus on defining a means to map the attributes of a service request against a set of available services using a Service Matching Algorithm.

3.1.1.1.3.5 Service Invocation

Service Invocation is concerned with actually instantiating a service between a provider and an authorized Requestor, subsequent to successful selection of Service Type and issuance of a Service Request. The invocation may be done via an API that is accessible to authorized consumers.

This effort will not develop technology for the invocation of services between producers and Requestors. The open API implementation to a Service Invocation System that will be utilized for this effort is the Sensis-developed Crystal Shield API.

3.1.1.1.3.6 Service Execution

Service Execution is concerned with providing the service to the authorized consumer according to the negotiated attributes. The enforcement of the service "contract" according to the negotiated attributes is the responsibility of a process that controls the stateful execution of the service contract according to the negotiated attributes.

This effort will not develop technology for the delivery of services, which is performed by a Service Invocation System.

3.1.1.1.3.7 Service Release

Service release is concerned with termination of service according to terms of successful completion, or violation of, the negotiated attributes of the granted service.

This Secure Service Manager developed in this effort will enable the de-activation of information services.

3.1.1.2 Review Current State-of-the-Art

Numerous services-based architectures have been defined, but very few have addressed security as a fundamental design consideration, and even fewer have considered end-to-end security as a core part of a services-based architecture. This subsection will discuss the security considerations of the most prominent architectures studied.

Figure 1 illustrates a protocol stack representing the current approach for addressing interoperability in a Web Services environment. We will differentiate this model with an Information Services environment in this project, which does not rely on Web Services or the Web Services protocol stack.

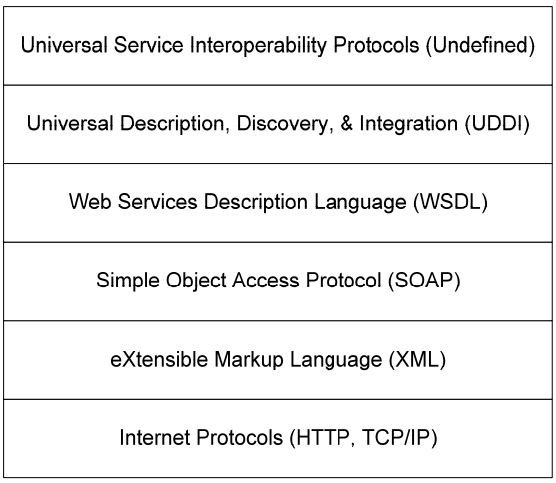


Figure 1: Web Services Interoperability Stack

Web Service vs. Information Service: A web service refers to specific business functionality that is exposed by a company for the purpose of providing a way for another company or software program to use the service over a network. We will focus our work on services that pertain to information sharing (information services), which are also able to be expressed and specified as web services, but do not depend upon a web services protocol stack for implementation.

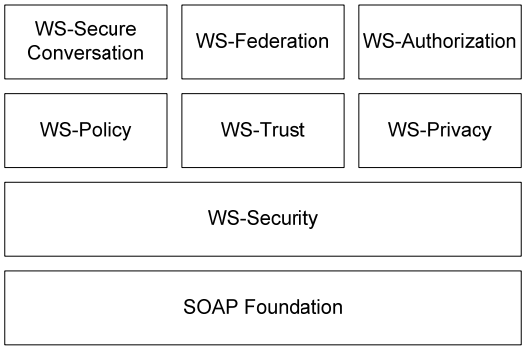
3.1.1.2.1 Security of Service Oriented Architectures

The US Department of Defense (DoD) is in the process of a massive migration toward Service Oriented Architectures (SOA) as part of their transformation to network-centric operations. The architectural benefits envisioned are decentralization, loose coupling, and improved interoperability. In a SOA, a set of network-accessible operations and associated resources are abstracted as a "service". The service is described in a standard (structured) fashion, published to a Service Registry, discovered and invoked by a service consumer. The emergence of SOA implementations has triggered a major paradigm shift in distributed computing.

The paradigm shift toward SOAs acknowledges the pervasive need for information exchange across organizational and network boundaries. A paradigm shift is also required in the security of services-based architectures, because most existing security approaches assume that information producers and consumers are on the same network. The security model must account for the fact that SOA participants may be in different organizations having different security policies. From a security perspective, information interoperability can not occur until security interoperability is established.

As stated in the DoD's Network-Centric Enterprise Services (NCES) Security Service overview: "in a Net-Centric environment, the focus on perimeter-based security models must be augmented with an application or service-level view of security. With both models in mind, the emphasis is placed not on physical ownership and control but on network identities, trust, and authorized access to resources by both users and other principals."[1]

Figure 2 shows the individual specifications that collectively comprise the Web Services Security Specification Framework.



[1] Quotation from http://ges.dod.mil/ServiceSecurity.htm

Figure 2: Web Services Security Specification Framework

The **WS-Security Specification**: WS-Security defines a standard set of Simple Object Access Protocol (SOAP) extensions. These message headers are enhancements to SOAP messaging to provide quality of protection through message integrity, message confidentiality, and single message authentication. WS-Security also provides a general-purpose mechanism for associating security tokens with messages, and how to encode binary security tokens. It also includes extensibility mechanisms that can be used to further describe the characteristics of the credentials that are included with a message. This specification provides an important foundation layer for developers to build more secure and broadly interoperable Web Services.

The **WS-Policy Specification**: WS-Policy defines the methods in which the capabilities and constraints of security policies can be expressed. It provides a grammar, framework, and model for expressing the capabilities, requirements, and general characteristics of entities in an XML Web services-based system as policies having one or more assertions that must be met.

The **WS-Trust Specification**: WS-Trust is a model for establishing both direct and brokered trust relationships. Figure 3[2] illustrates a graphical representation of the WS-Trust model.

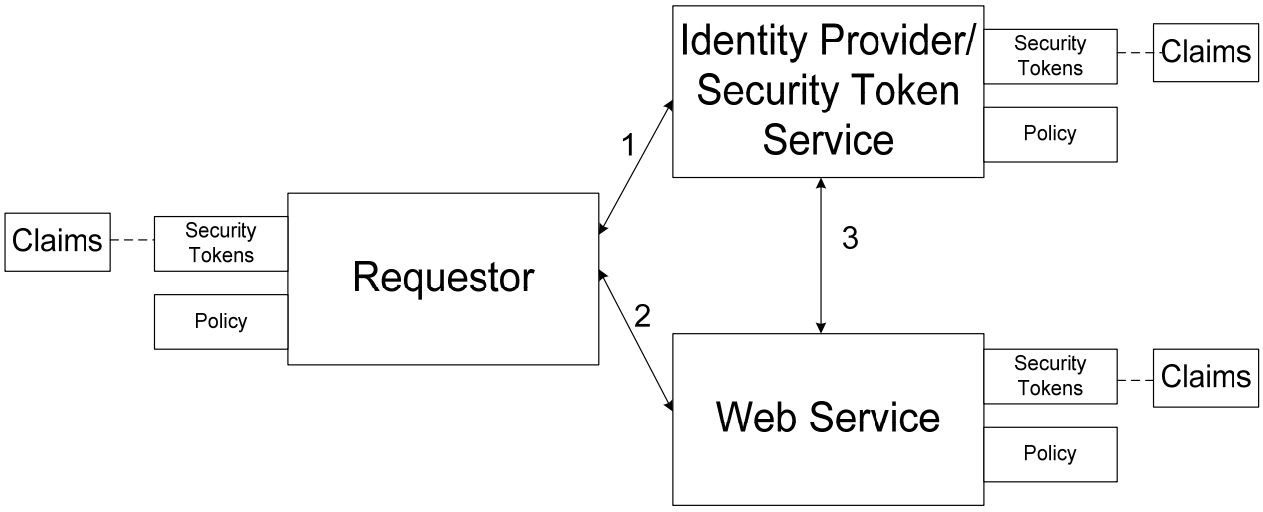


Figure 3: Web Services Trust Model

The **WS-Privacy Specification**: WS-Privacy is a specification that addresses how privacy practices can be stated and implemented by Web Services.

---

[2] "Security in a Web Services World: A Proposed Architecture and Roadmap", A Joint White Paper from IBM Corporation and Microsoft Corporation, April 7,2002 (Version 1.0)

The **WS-Secure Conversation Specification**: WS-Secure Conversation describes how message exchanges can be securely managed. It also deals with security context exchange and establishing and deriving session keys.

The **WS-Federation Specification**: WS-Federation relates to managing and brokering trust relationships in a heterogeneous distributed environment. It also includes support for distributed computing.

The **WS-Authorization Specification**: WS-Authorization is a standard for how to manage authorization data and policies for Web Services.

Figures 4 through 9[3] illustrate 6 different examples of how trust can be brokered using the WS-Fed model.

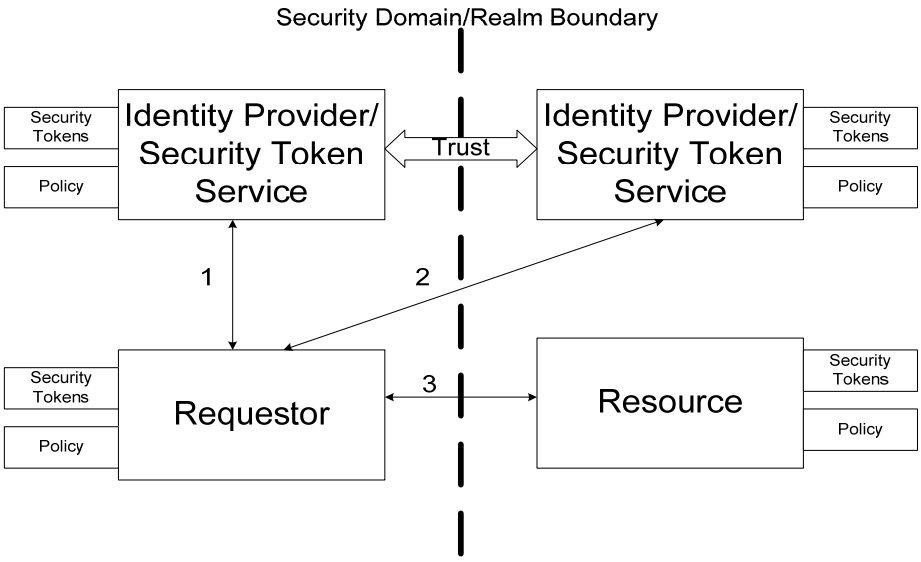


Figure 4: Web Services Federation Trust Model Example 1

[3] Web Services Federation Language (WS-Federation), IBM, Microsoft, BEA Systems, RSA Security, Verisign, Version 1.0, July 8, 2003.

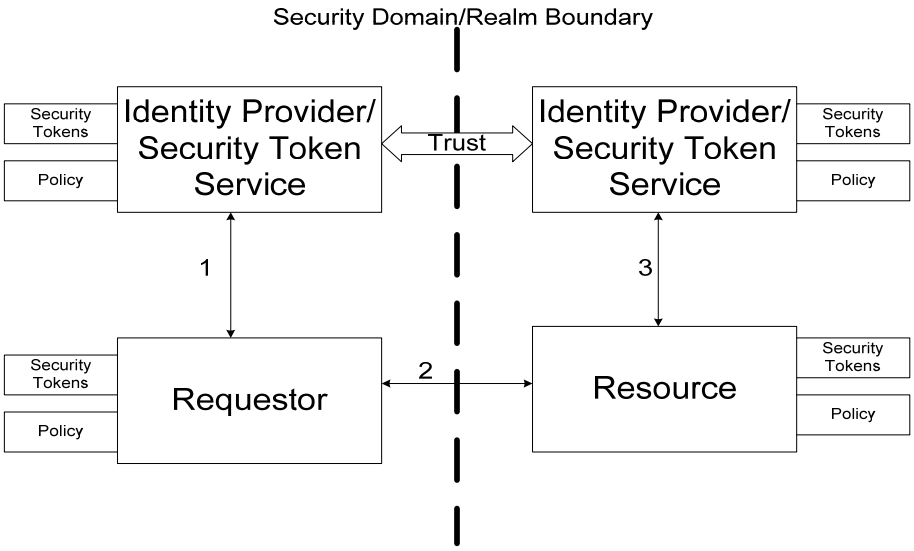


Figure 5: Web Services Federation Trust Model Example 2

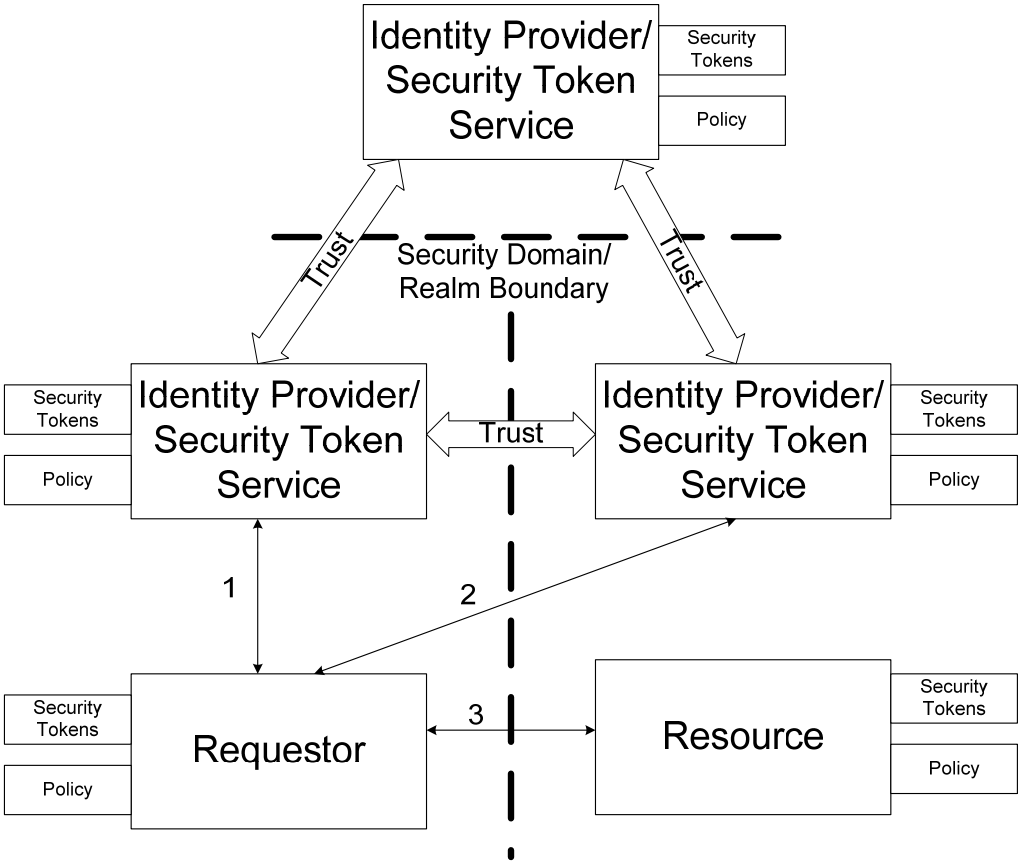


Figure 6: Web Services Federation Trust Model Example 3

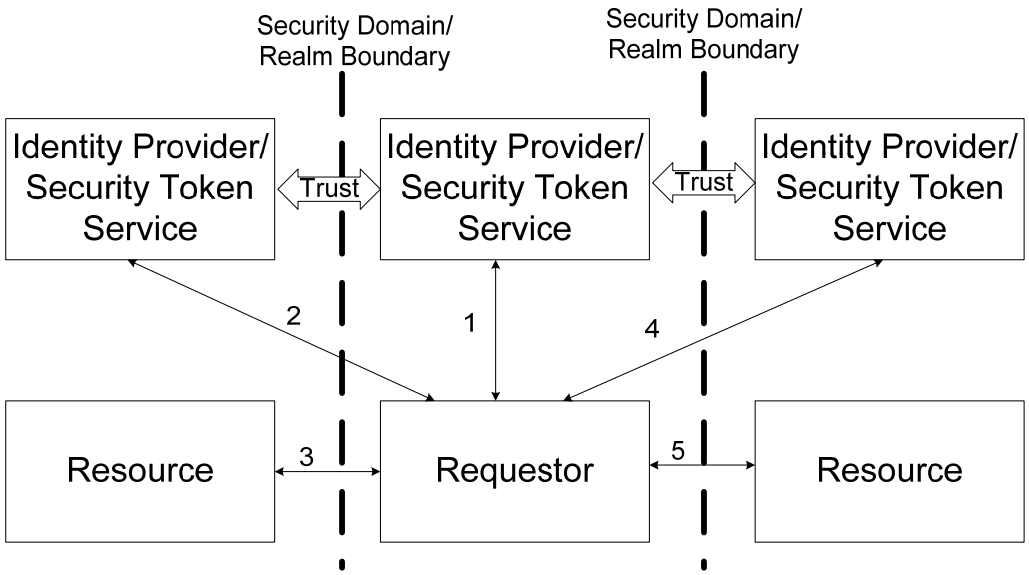


Figure 7: Web Services Federation Trust Model Example 4

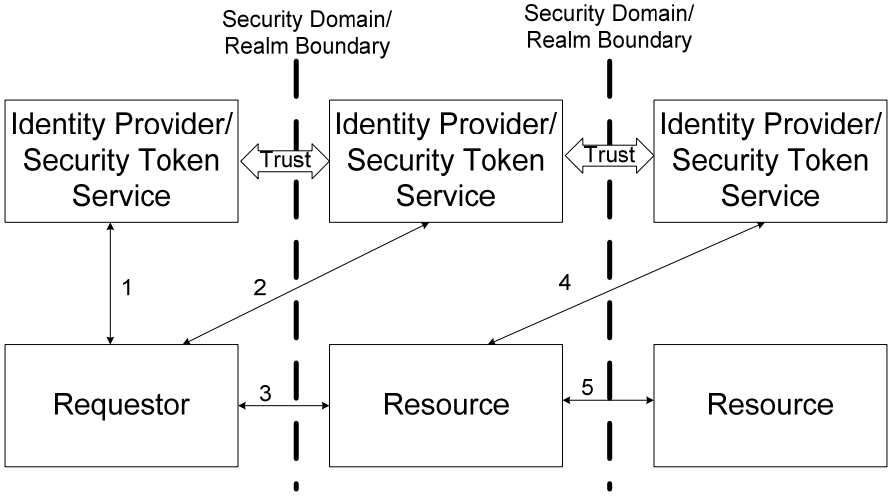


Figure 8: Web Services Federation Trust Model Example 5

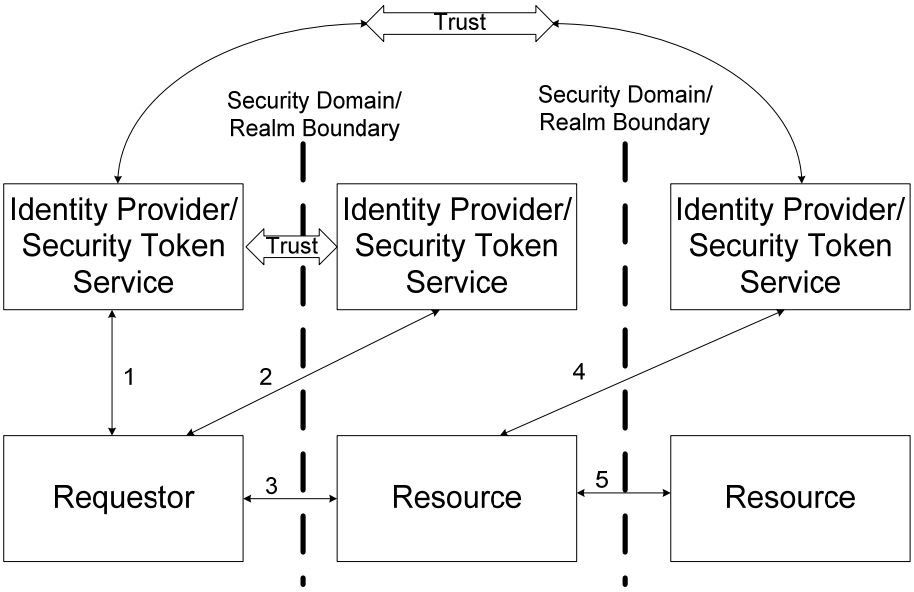


Figure 9: Web Services Federation Trust Model Example 6

This project is not based upon Web Services in its design, but uses the preceding discussion of the Web Services Security Specifications and Security Models to illustrate the specific security-related considerations that are widely in use in current SOA architectures. Our research identifies the weaknesses in these security models, which forms the basis for our recommended security architecture for information Service Management.

3.1.1.2.1.1 Service Registries

Service providers publish (announce) their service description (specification) to a registry. Service consumers go to registries at known locations to "discover" what services are available to them. Using the Web Services framework and associated security model described above, Service Registries provide location of and a means to bind to service providers. The most common service registry model is that for UDDI (Universal Description, Discovery, and Integration). A UDDI registry may be implemented as a public registry that is accessible to all Requestors regardless of their domain, or as a private registry that makes its content selectively available to authorized external Requestors. Figure 10 illustrates a basic information model for a UDDI Registry.

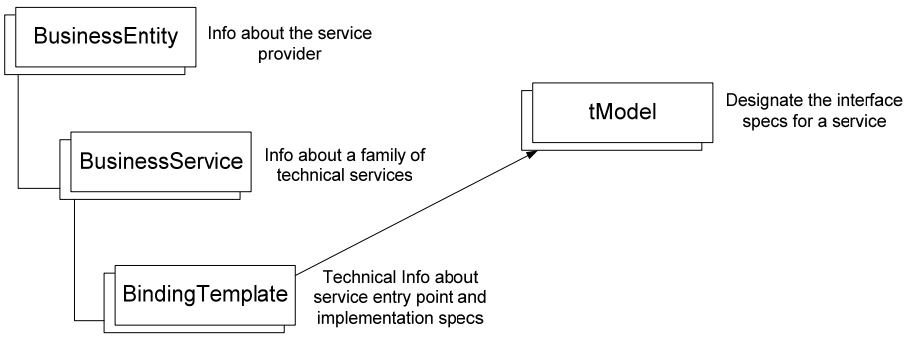


Figure 10: Basic UDDI Registry Information Model

Figure 11 illustrates an example Private UDDI Registry Trust Model implemented across two Security Domains.

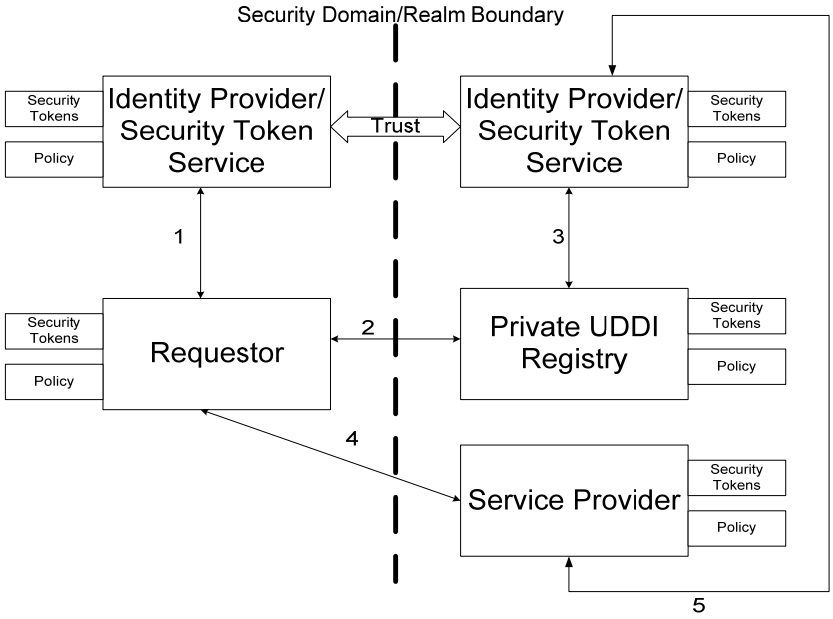


Figure 11: Private UDDI Registry Trust Model

A security concern in both Figures 10 and 11 is the fact that the Binding Template of the UDDI Registry provides the direct location information regarding the service provider. We assert that this knowledge of service provider location exposes a security vulnerability that may be exploited to gain access to unauthorized information, or other security compromise of the service provider.

From a security perspective, we propose a secure private Service Registry, where the contents of each collaborating domain's local registry are selectively shared only with trusted domains. Only services that are to be exposed to Requestors in external domains are shared outside of a local Security Domain. Rather than give external Requestors access to a local Service Registry across a Security Domain boundary, the local domain's Secure Service Manager is responsible for distributing the Service Announcements that are to be shared with trusted external domains, and each corresponding external domain's Secure Service Manager must decrypt the Service Announcement and update their local registry. Our approach contributes a unique method for securely encrypting and decrypting cross-domain Service Announcements in a manner that prevents any unauthorized entity from compromising the security of the Service Announcement method.

The local domain Service Registries will not contain location information of the service providers, because this is "virtualized" by each domain's Secure Service Manager for security purposes, although metadata will be available allowing the Requestor to know that the provider is not a local service to aid in selection and negotiation decisions. When a Requestor decides to invoke a service, the local domain's Secure Service Manager forwards the Service Invocation Request to the Service Invocation System in its local domain, which invokes the service if it is local, or securely forwards the Invocation Request to the appropriate external domain's Service Invocation System if the service provider is not local. The local domain's Secure Service Manager maintains a stateful awareness of the status of Service Requests that originated in its domain.

3.1.1.3 Core Requirements for Security of Information Services

In a multi-domain environment in which Information Services must be shared both within and across Security Domain boundaries, a security framework should include all the basic security requirements addressed in the WS-Security Specifications, AND location transparency of both Service Registries and service providers across Security Domain boundaries. The contribution of this effort is to define a comprehensive framework that includes all of these security considerations, to eliminate the vulnerabilities of current approaches.

3.1.1.4 Security Gaps in Current Technology

This subsection summarizes the (security) vulnerabilities of current approaches for sharing information services across Security Domain boundaries.

In current Service Registries, location information of the service provider is directly available in the Service Announcement in the Registry. Even if the registry information were access-controlled between security domains, all authorized Requestors (and un-authorized requestors who may find a means to circumvent the access control) would have direct access to the location of service providers.

Exposing direct location information of a service provider within a Service Registry is a vulnerability both within a security domain (insider attacks), and especially across domain boundaries.

Exposing direct location information of a Service Registry itself is also a security vulnerability, especially if the Registry is to be accessible to Requestors from outside the local Security Domain.

Our assertion is that current approaches (and architectures) have addressed overall information security in a manner that still leaves significant vulnerabilities in system security.

Trust in current architectures is more decentralized, but still open to exploitation of:

- Location of service providers.
- Location of service registries.

Misuse of this "free" information can lead to direct attacks (both internal and external) that exploit vulnerabilities in the underlying network and host operating systems of any participating domain.

3.1.1.5 New Model for Automated Secure Management of Information Services

Security Domains choose to collaborate based upon a high degree of trust established between the participating domains, ensuring that any participating domain is not compromised because of the collaboration (made vulnerable to malicious theft of data or attack on the system and its resources). Establishing a high confidence of trust requires that security vulnerabilities associated with cross-domain communication be eliminated to the greatest extent possible.

The "management" of an information service is comprised of Service Announcement, Discovery, Negotiation, Request for Invocation, and Termination. Details regarding the instantiation of a service (i.e. - the steps involved in activating a service flow) between a producer and consumer are out of scope of this effort, although an open API is provided to a generalized external Service Invocation System. This effort is focused on defining a new model for management of information services across security boundaries, in a manner that is consistent with the following core security principle centered on preventing the compromise of a security domain:

Location information of each domain's internal hosts, processes, services, and data is not revealed across security domain boundaries. This prevents vulnerabilities associated with internal attacks, as well as direct attacks from outside the local domain. This accounts for many of the attacks presently associated with today's systems.

This security requirement imposes the following new constraints upon the management of information services between collaborating Security Domains:

- Trust is placed in a domain, not in individual users. Domains verify their unique identities to each other preceding any form of inter-domain collaboration.
- Service Registries must be distributed, and Service Specifications must be location-independent. A Service Registry will know what services are provided by its internal

providers, and its trusted external domains, but not what the directly addressable locations are for any of those providers. A local domain will only know the location of each trusted external domain to which it can communicate.

Each local domain must be responsible for securely sharing (announcing) the local services that it is willing to provide to authorized external domains. This information must be available to each trusted domain, which is responsible for controlling access to it accordingly, in its local registry.

- The process of service "discovery" in the secure case is reduced to a secure, fast look-up (Service Type selection) in the local domain of all services to which a consumer has access, which already includes all external services that have been announced between trusted domains. Trusted domains update their Service Announcements to each other whenever there is a change to a local domain service (add, modify, delete).

- Service Negotiation and selection (matching) is a process of determining which individual instance(s) of an announced Service Type best satisfies a Requestor's needs. There are 2 basic models for how the consumer's needs may be fulfilled (though this effort is only concerned with the latter model because it offers more flexibility and a richer capability):

    - Consumer visually inspects the registry contents via a GUI, manually searches for matches based on the advertised service descriptions, and selects a service having the most desirable attribute values. The registry provides an API to which the consumer may use to "bind" their application to the advertised service.

    - Consumer generates a template having the desired Service Attributes required to satisfy their information needs, along with a set of preferred "Weights" assigned to the relative importance of each attribute. (We assume that those needs are expressed as a set of attributes corresponding to a Service Type.) The template is issued to the Secure Service Manager's API, and it is the responsibility of an internal software process to perform a "matching" operation against the contents of the registry. The possible outcomes are:

        - There exist one or more "exact matches" against advertised services in the registry that contain all the Requestor's desired Service Attributes.

        - There exist only partial matches against service instances that are in the Registry.

        - It is possible that no provider satisfies the required attributes of the request.

3.1.2 Recommended System Architecture

Figure 12 is an illustration of the recommended architecture for the Secure Service Management (SSM) system, which is labeled as a "Domain Service Manager" to denote different instances of an SSM in each Security Domain.

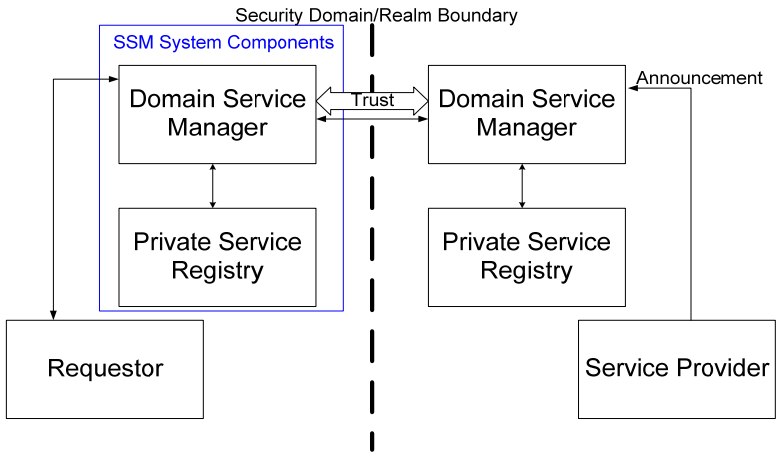


Figure 12: Recommended Service Management System Architecture

Figure 13 illustrates the recommended SSM System along with a Service Invocation System.

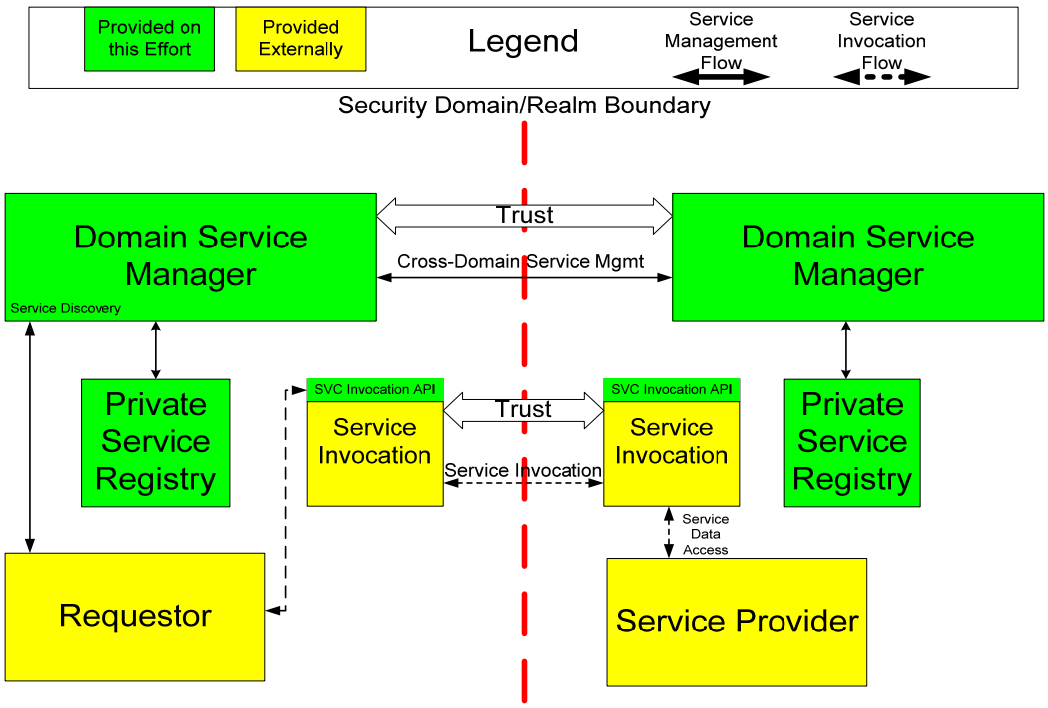


Figure 13: Service Management Architecture with Service Invocation Subsystem

3.2 System Interface Requirements

3.2.1 External Interfaces

3.2.1.1 SSM Interface to Local Domain Requestors

SSM provides an interface to Requestors in its local Domain which allows Requestors to:

- Select an Available Service Type
- Receive a Service Type Template
- Select Service Attributes and Assign Weighted Attribute Preferences
- Discover Available Services using Weighted Preferences
- De-Activate a Service

3.2.1.2 SSM Inter-Domain Interface to External SSM

SSM provides a point-to-point interface to each external SSM in other trusted Domains to which it must communicate during the Service Announcement process.

3.2.1.3 SSM Administrative Interface

SSM provides a means to administratively enter Service Specifications into the SSM's local Domain Service Registry.

3.2.1.4 Requestor Interface to a Service Invocation System

For posting a Service Invocation Request with a single target item, the Requestor's application **will** {3.2.1.4-1} utilize the following open interface specification provided by a Service Invocation System:

RequestId PostRequest(

in SessionID auth_token

, in string callback_IOR

, in UserIdentity user_id

, in Request request_);

Where:

auth_token = Kerberos authentication token

callback_IOR = IOR of RequestCallback object (location of Requestor’s callback interface)

user_id = Requestor’s client identity

request_ = the single Request item

(<Requestor> requests operation <R,W,P,S> on <resource>)

Returns RequestID for valid Request

For posting a Service Invocation Request with a multiple-target item, the SSM **will** {3.2.1.4-2} conform to the following open interface specification provided by a Service Invocation System:

RequestId PostRequests(

in SessionID auth_token

, in string callback_IOR

, in UserIdentity user_id

, in RequestList requests);

3.2.1.5 SSM Interface to a Domain Administrator

The SSM **will** {3.2.1.5-1} provide a means for a Domain Administrator to enter Service Specifications (Schemas for the Service Types defined in a local Domain) into the SSM's local Service Registry.

Requirement 3.5.2.1-4 describes the interface provided by an SSM to a Domain Administrator for initializing the entries for trusted external SSMs and their location in the local Shared Services Domain Directory.

3.3 System Functional Architecture

This section describes the overall SSM functional architecture, and the concept of execution among the system functions.

3.3.1 System Functional Description

Table 1 lists the SSM system functions and summarizes their responsibilities and major subfunctions.

Table 1: SSM System Functions

System Function	Description/Subfunctions
Service Announcement	• Securely announce information services across Security Domain boundaries between two trusted Domains
Service Registration	• Push advertised information services (including local and external Domain Service Announcements) into a secure local Repository.
Service Repository	• Securely store information Service Specifications for defined Service Types. • Securely store information Service Announcements.
Shared Services Domain Directory (SSDD)	• Secure Domain-internal storage for identification of which Service Types & Instances a Domain will share with trusted external Domains, along with location information for the SSM of each external Domain in the SSDD.
Service Discovery	• Provides an interactive interface to Requestors for: o Secure lookup of the Service Types available. o Selection of Service Type desired. o Entry of desired Service Attribute values, or range. o Assignment of Weighted Preference of Attributes. o Submittal of an Initial Service Request.
Service Request Matching	• Processes the Requestor's Initial Service Request against the content of Service Repository and determines the relative ranking of matching results in accordance with the Weighted Preference of Service Attributes desired by the Requestor.
Service Release	• Provides the Requestor a means to de-activate a Service.

3.3.2 Functional Interaction During Service Announcement

Figure 14 illustrates the time-sequenced flows among the SSM functions that participate in the Service Announcement process.

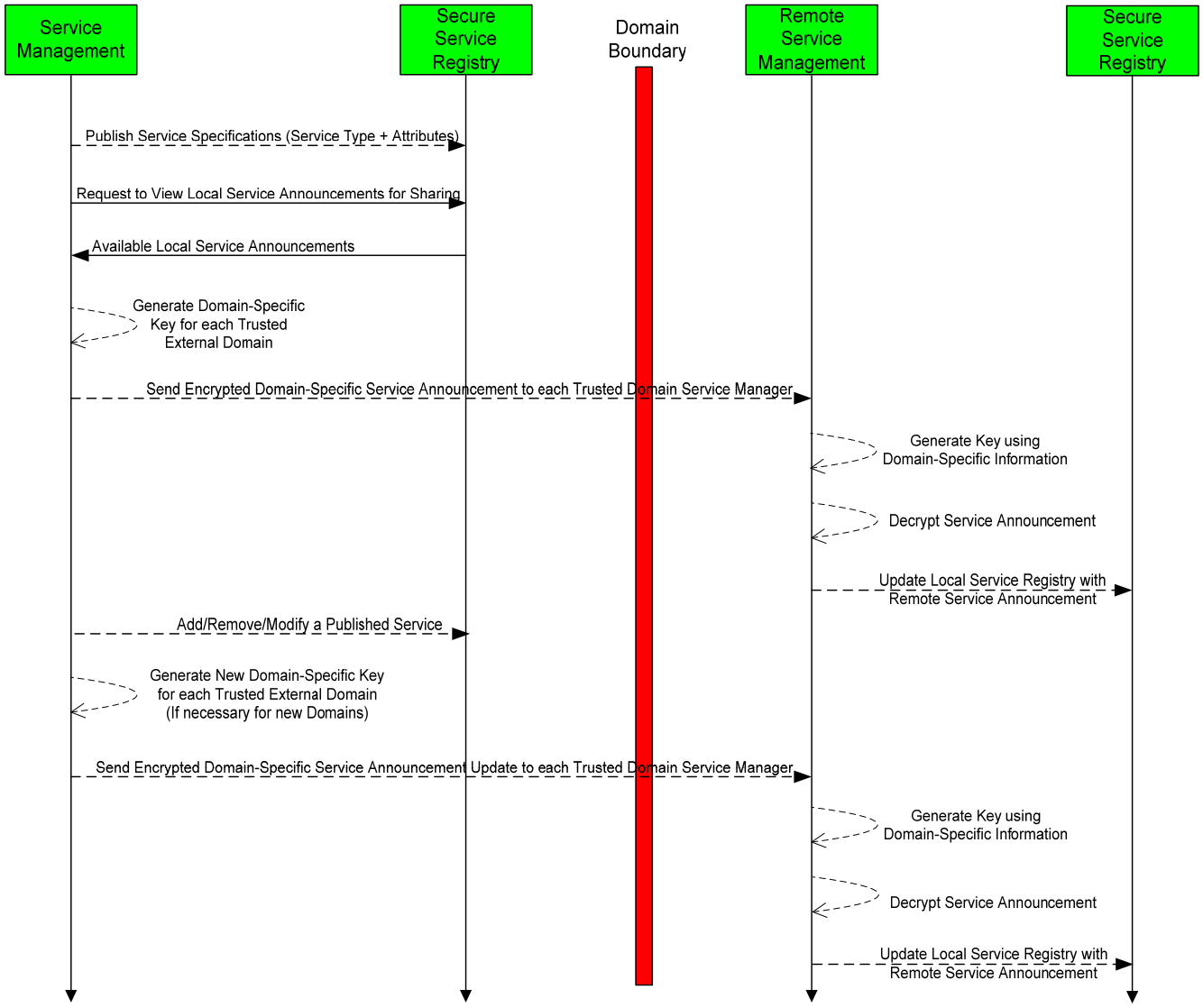


Figure 14: Secure Multi-Domain Service Announcement Process

3.3.3 Functional Interaction During Service Management Cycle

Figure 15 illustrates the time-sequenced functional interaction between SSM processes that interact during the Service Management cycle.

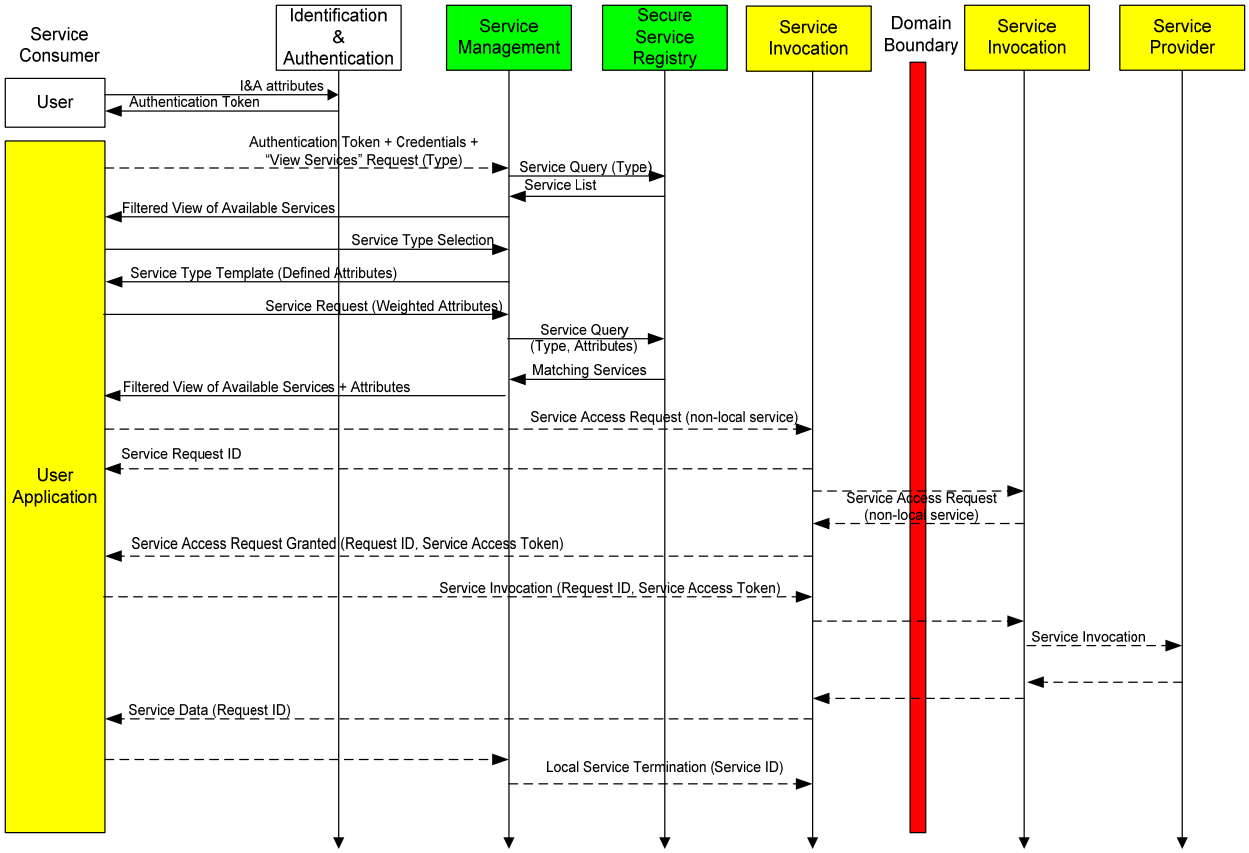


Figure 15: Secure Information Service Management Request Processing Cycle

3.4 System Physical Architecture

This section describes the physical system architectural design of the SSM, including a block diagram of the system. SSM is comprised of a single CSCI. There is no HWCI for the SSM system.

3.4.1 System Physical Description

A diagram of a typical top level physical system that uses SSM in a single security Domain is shown in Figure 16, with the SSM components shown in red. All SSM processes are hosted on a single computer which is on a TCP-IP based network. On the same physical network are Requestors (and their applications), a Service Invocation System (on one or more hosts that implement Service Invocation processes), and all networked service providers.

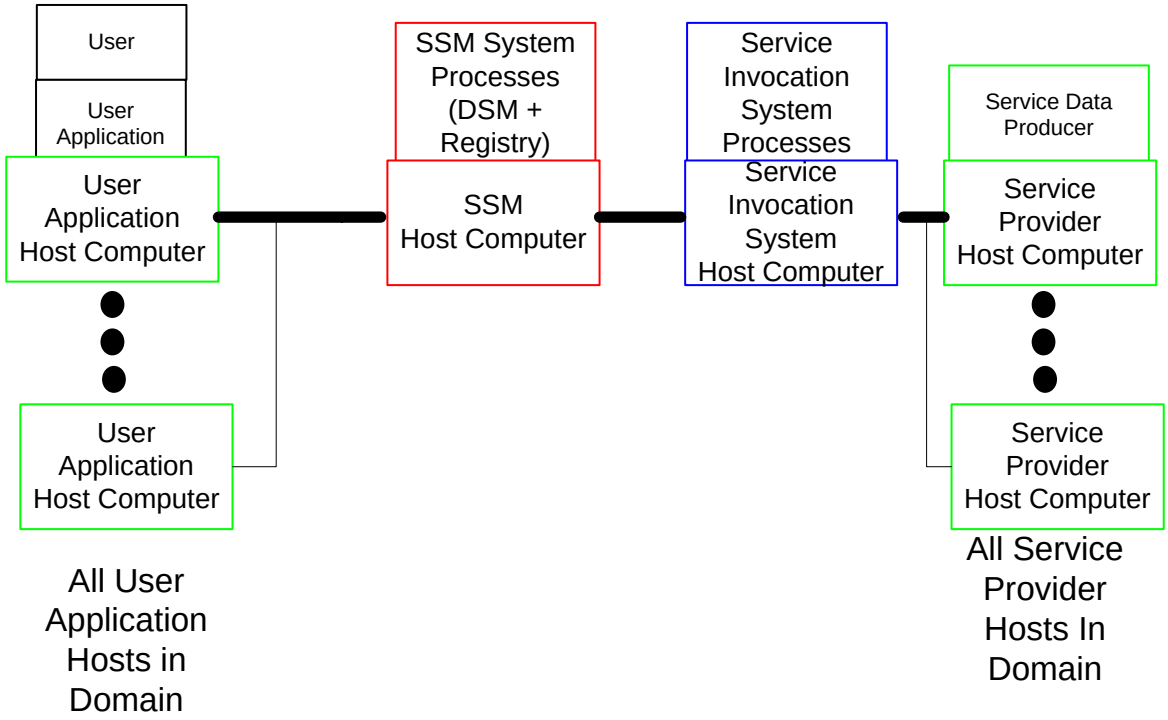


Figure 16: SSM System Physical Architecture

3.4.2 System Internal/External Interface Description

The SSM external (functional) interface architecture is described in Sections 3.2. The external (physical) interface architecture is a standard TCP/IP networked host computer. The SSM internal interface architecture was illustrated in Figures 14 and 15, and described in Section 3.3.1.

3.4.3 System Internal Data Description

Internal to the SSM CSCI, between the core functional components, the component-to-component messages are all Sensis-defined and are specified in the Software Design Document[4].

3.5 CSCI Requirements

This section specifies the Computer Software Configuration Item (CSCI) requirements for the Secure Service Management (SSM) System, which capture the characteristics of the system that are the conditions for its acceptance. The SSM System is composed of a single CSCI. The

[4] Composable Infospaces Software Design Document, Sensis Corp., September 2006, Document #710-014708.

CSCI requirements are organized into groups of system capabilities, as specified in Subsection 3.5.1.

3.5.1 Capability Requirements

This subsection organizes CSCI requirements into groups of capabilities.

3.5.1.1 Secure Information Service Announcement

Cross-Domain communication for Information Service Announcements **will** {3.5.1.1-1} occur only in a point-to-point manner between pairs of SSMs that must share Services.

An SSM **will** {3.5.1.1-2} be capable of sharing only its ***local*** Services with the SSM of external Domains.

Each SSM **will** {3.5.1.1-3} generate and maintain a secure internal Shared Services Domain Directory that identifies the specific Information Service types and instances that it will share with each trusted external Security Domain, and the location information for the SSM of each external Domain in the Directory.

The entries in the Shared Services Domain Directory **will** {3.5.1.1-4} be editable only in the local Domain, either through a configuration file, or directly through an editor interface.

An Information Service Announcement is specific to the trusted external Domain to which the local Domain wants to share one or more of its local Services. Therefore, there is one Information Service Announcement generated by the local SSM for each external Domain to which services must be shared.

Each SSM **will** {3.5.1.1-5} generate an Information Service Announcement for each trusted external Domain in its Shared Services Domain Directory.

An Information Service Announcement **will** {3.5.1.1-6} contain all the following:

- The Information Service Type(s) to be shared with a specified trusted external Domain.
- The Information Service Specification (schema) for each Service Type that is shared.
- The instances of announced Services for each Service Type that is shared.

Each SSM **will** {3.5.1.1-7} protect all of its outgoing Information Service Announcement(s) in accordance with the following encryption methodology:

- The local SSM generates a Domain-Specific Key that is unique to each trusted external Domain listed in the local Shared Services Domain Directory, which contains domain-specific information that only the intended recipient Domain's SSM can verify in order to decrypt this Key.
- The local SSM encrypts each outgoing Service Announcement with a Domain-Specific Key for the intended recipient Domain.

- The local SSM transmits each outgoing encrypted Service Announcement to the SSM of the specific external Domain to which it intends to share the selected set of Services in the Announcement.
- Each external SSM that received an encrypted Service Announcement from the local Domain generates a Key with the Domain-Specific information that only it can provide.
- The recipient external Domain decrypts the received Service Announcement with the domain-specific key it generated.

Timing of Service Announcement transmittal is driven entirely by each local Domain, and occurs as a selective "push" of Service Announcements to trusted external Domains.

Each SSM **will** {3.5.1.1-8} transmit its outgoing Information Service Announcement(s) to each intended trusted external Domain in the Shared Services Domain Directory in accordance with the following timing constraints:

- Upon SSM initialization.
- Upon every update to the local Service Registry, which occurs:
    - Upon addition/removal of a Service Type.
    - Upon addition/removal of an ***instance*** of an active service type (i.e. - individual instances of Services announced by service providers).
- Upon addition of a new entry in the local Shared Services Domain Directory.

For each instance of an announced Service available in a local Domain, the local SSM **will** {3.5.1.1-9} maintain secure internal knowledge of the location (address) of the SSM that announced the Service.

The following summarizes the approach utilized for Domain-specific encryption of service announcements:

- Keys are generated dynamically and are based on the individual domains involved in the service announcement.
- Keys are never communicated between domains, reducing the likelihood of key discovery by attackers.
- Keys are generated when needed, not stored, at the domain performing the encryption/decryption, also reducing the likelihood of key discovery by attackers.
- Service announcement keys are generated based on information from BOTH machines involved in the announcement.
- For maximum security, machine information is communicated out-of-band to the remote machine involved in the trust relationship.
- Information used to dynamically generate keys is partially based on machine-specific information.

- Machine-specific information from both domains is combined and reordered before being used to generate the key.
    - Local machine information is retrieved at time of key generation and remote machine information is communicated out-of-band.
        - Release
        - Version
        - IP Address
        - Port
    - Portions of version information and port are used in key generation

The following summarizes the methodology for key generation:

- Versions for each machine have the time extracted as a 6-digit string.
    - #2 SMP Tue Mar 7 08:19:39 UTC 2006 has 081939 extracted.
    - Time is used because it is likely to be unique between domains.
- Version information is combined to form a 24-byte string as follows (assume a remote domain extraction of 134629):
    - A twelve-byte string is created by concatenating the two extractions: 081939134629.
    - Then concatenating the result with itself: 081939134629081939134629.
- First port number is decomposed into its component bits (16 bits).
- Component bits are split into two sets of 8 bits each (PA1 & PA2).
- Second port number is decomposed and split into 2 8 bit sets (PB1 & PB2).
- An array of 8 bytes is created using the 4 sets of 8 bits in the order PA2 PB1 PA1 PB2 PB2 PB1 PA2 PA1.
- A String based on version information is used as the password.
- An array derived from the port information is used as the salt.
- Key is generated and used to encrypt or decrypt information for/from remote machine.
- Key generation is part of the encryption and decryption methods – keys are never retrieved from any type of storage, they are generated "on-demand."

3.5.1.2 Secure Information Service Local Registration

A local SSM **will** {3.5.1.2-1} maintain a secure local Service Registry that dynamically keeps track of the following entries:

- A Service Specification (Schema) for each Service Type registered by the local SSM.
- An entry for each instance of a Service Provider's Service Announcement that describes the available Service, including its attribute values.

The content of the local Service Registry **will** {3.5.1.2-2} be accessible only by its local SSM, not directly to Domain users.

The content of the local Service Registry **will** {3.5.1.2-3} be stored encrypted.

All non-numeric information stored in the Service Registry database is encrypted, including:

- Trusted domain information
- Service information

This methodology provides protection against an attacker dumping the database and acquiring sensitive information. The key utilized for Registry encryption is different key than the key used for service announcement encryption. Encrypted data is base 64 encoded before being stored.

3.5.1.3 Secure Information Service Discovery

3.5.1.3.1 Service Type Selection

A local SSM **will** {3.5.1.3.1-1} provide an interface to local Domain Requestors (where a Requestor may be a user via an interactive display interface, or a user's application process) that allows the Requestor to select a desired Service Type from the set of Service Types available in the Domain.

Upon selection of a desired Service Type via the Service Type Selection interface, the local SSM **will** {3.5.1.3.1-2} provide the Requestor with a Service Type Template containing the attributes defined in the Service Specification for the chosen Service Type.

3.5.1.3.2 Service Attribute Selection and Weighted Preference

Within the Service Type Template containing the Service attributes, the local SSM **will** {3.5.1.3.2-1} provide the Requestor with a means to assign their desired value, or a range of values, for each Service attribute in the Service Type Template.

The local SSM **will** {3.5.1.3.2-2} provide the Requestor with a means to assign their choice of the following Weighted Attribute Criteria for each Service attribute value, or range of values, that they specified in the Service Type Template:

- An assignment for whether the Requestor-provided attribute value is "Mandatory", "Preferred", or "Not Required".
- For "Preferred" attributes, an additional option to assign a preferred "Weight" to the attribute, in the form of a percentage (where Mandatory denotes an implied 100%, "Not

Required" denotes and implied 0%, and "Preferred" may be a Requestor-assigned Weight having a value between 0% and 100%).

3.5.1.3.3 Submittal of Initial Service Request

Along with the Service Type Template containing the Service Attributes, the SSM **will** {3.5.1.3.3-1} provide the Requestor an option to submit their final Weighted Attribute Criteria for the Service Type Template as an Initial Service Request to the SSM.

3.5.1.4 Initial Service Request Negotiation

An Initial Service Request denotes what a Requestor desires, in terms of the Mandatory and Preferred Service Attributes, and their values and preferential weights. Once an Initial Service Request is received by a local SSM, the SSM must determine if there are active instances of available Services that have attribute values that "match" those in the Initial Service Request.

The SSM **will** {3.5.1.4-1} implement an Service Request Matching Algorithm that processes the content of the local Service Registry and determines if there are active instances of available Services having attribute values that "match" those in the Requestor's Initial Service Request.

3.5.1.4.1 Identification of Matches and Ranking of Results

The SSM **will** {3.5.1.4.1-1} rank the results of its Request matching operation in order from highest to lowest, where the highest possible level of match denotes that all Mandatory and Preferred attributes specified in the Initial Service Request were present in at least one instance of an active Service.

The SSM **will** {3.5.1.4.1-2} assign a Ranking Value to each Service instance in the matching result that denotes a mathematical ratio of that match's set of attribute values relative to the set of all desired attribute Weights specified in the Initial Service Request.

In the case where there are multiple matches having the same Ranking Value, the Matching Algorithm **will** {3.5.1.4.1-3} order them sequentially in the result set with clear distinction that they are equivalent in Ranking Value.

Prior to returning the results of its Service Request Matching Algorithm, the SSM **will** {3.5.1.4.1-4} provide the Requestor the number of matches in the result along with the option of receiving either:

- A list of all matching Service instances
- A Requestor-defined number of the highest-ranked matching Service instances

If one or more of the Mandatory attributes can not be found by the Matching Algorithm in any instance of all the active Services, this **will** {3.5.1.4.1-5} be signified as a "no-matches found" condition.

In the case of a "no-matches found" condition, but there were instances of active Services having one or more of the Preferred attributes present, this **will** {3.5.1.4.1-6} be signified as a "partial matches available" condition, and the Requestor shall be provided the option to view the closest partial matches that were found in priority order.

In the case of a "no-matches found" condition, and there were no instances of active Services having one or more of the Preferred attributes present, the SSM **will** {3.5.1.4.1-7} inform the Requestor of the result, and provide an option to change the attribute Weights and submit a new Initial Service Request.

3.5.1.5 Service Release (De-Activation)

The SSM **will** {3.5.1.5-1} provide the Requestor a means to release (de-activate) a Service.

3.6 Software Detailed Design

The detailed software design for the SSM Concept Demonstrator system is available in the Software Design Document delivered to AFRL on this contract (Sensis Document Number 710-014708).

3.7 Proof-of-Concept Demonstration

3.7.1 Demonstration Scenario

The concept demonstration is comprised of two (2) Security Domains: a Federal Aviation Administration (FAA) Domain and a Department of Defense (DoD) Domain. The security policies of each Domain must be enforced for secure collaboration & data sharing.

The scenario is focused on a collaborative Community Of Interest (COI) Business Process: the shared FAA/DoD responsibility for coordinated establishment, safe utilization, and de-activation of Military Special Use Airspace (SUA) Ranges.

The collaborative workflow between the two Domains is based upon a “SUA Coordination Service” (See Figure 17) which is utilized to enable the following collaboration:

- DoD proposes required military SUA Region(s).
- FAA must re-allocate airspace to divert commercial flights around SUA region, and “clear” proposed regions for safe military use.
- DoD activates use of a "cleared" SUA Region.
- Selective data sharing via a “Radar Site Data Service” (See Figure 18) occurs, whereby situational awareness (data from individual radar sites) inside SUAs can be shared or restricted depending on policy, both locally inside the DoD Domain, and externally with the FAA Domain.
- DoD de-activates use of SUA Region.

- FAA regains airspace for commercial use.

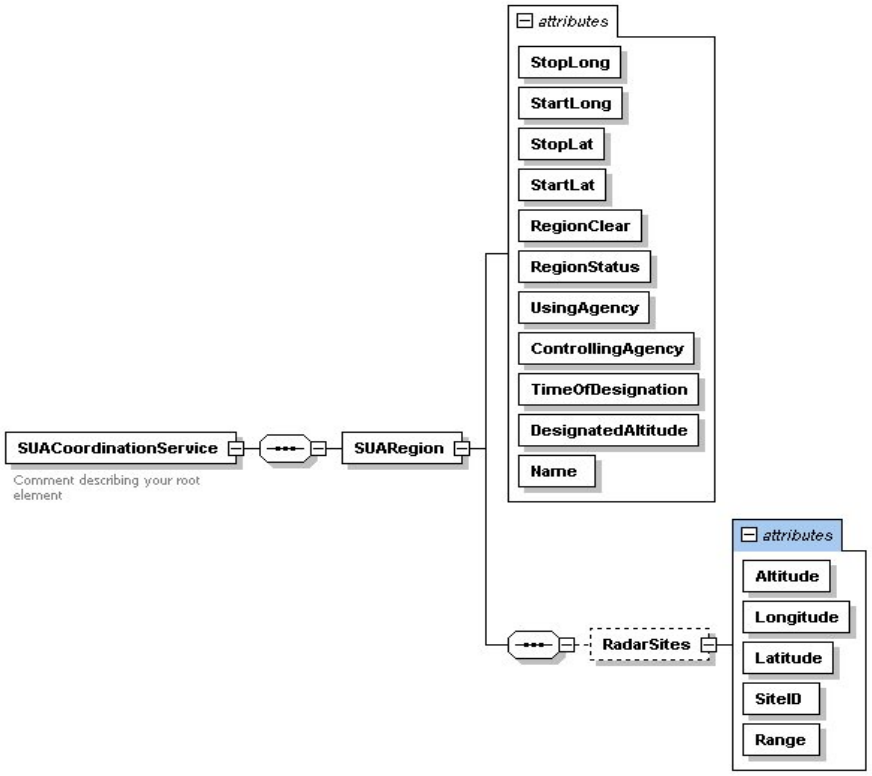


Figure 17: SUA Coordination Service Schema

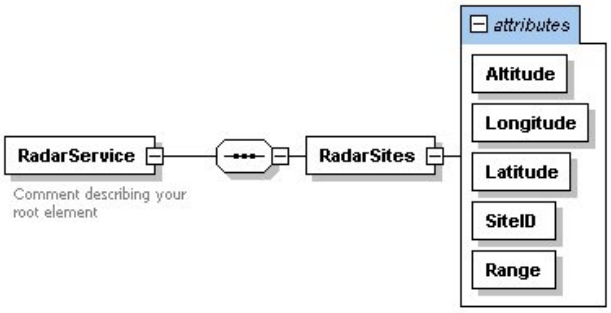


Figure 18: Radar Site Data Service Schema

3.7.1.1 Demonstration Scenario Script

Figure 19 illustrates the 2-Domain configuration that was utilized for the Concept Demonstration.

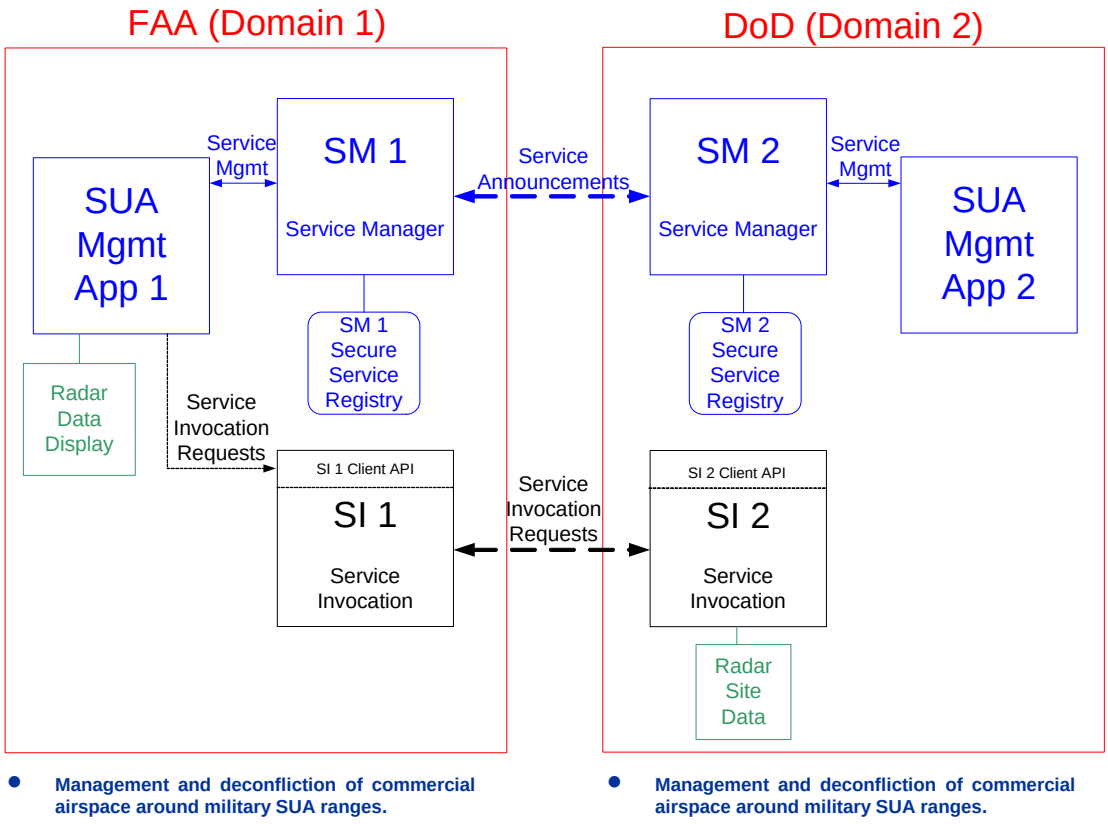


Figure 19: Concept Demonstrator System Configuration

Figure 20 illustrates the sequence of steps that are performed by all the components participating in the FAA and DoD Domains during the demonstration scenario.

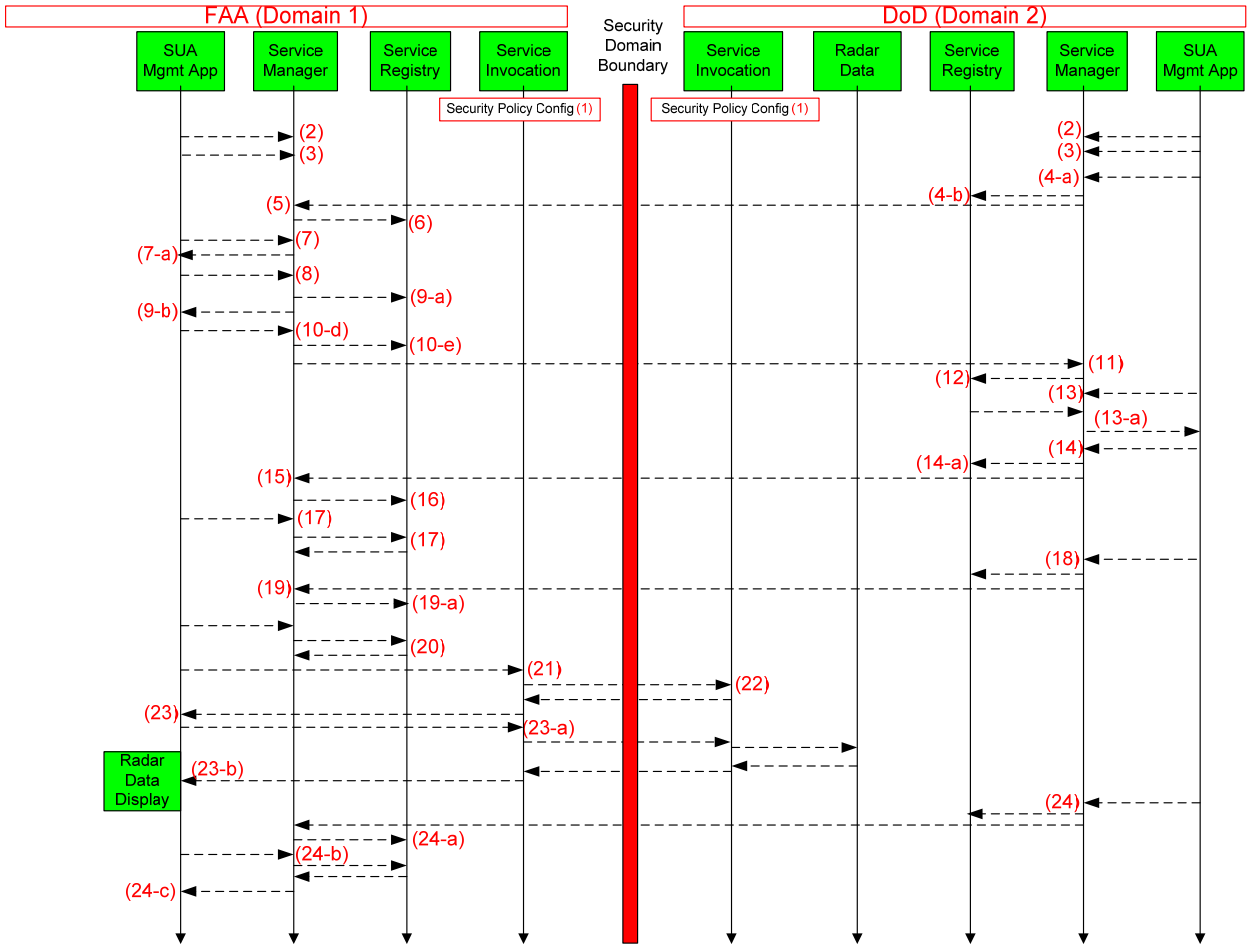


Figure 20: Concept Demonstration Sequence Diagram

3.7.1.1.1 Summary of Demonstration Scenario Steps

The following list summarizes the primary activities that take place in the step numbers shown (in red) on the sequence diagram in Figure 20.

Step 1- Configuration of FAA & DoD Security Policies.

Steps 2, 3- Define Service Types & Trusted Domains

Steps 4, 5, 6 - DoD announces new SUA Region 1 via SUA Coordination Service, Status =“Proposed”.

Steps 7, 8, 9- FAA discovers new SUA Coordination Services that match geographic region of responsibility for SUA Manager.

Steps 10, 11, 12- FAA announces “Region 1 Clear = Yes” via SUA Coordination Service.

Step 13- DoD discovers SUA Coordination Services having “Region Status = Proposed” & “Region Clear = Yes”.

Steps 14, 15, 16- DoD announces “Region 1 Status = Active” via SUA Coord Service.

Step 17- FAA discovers SUA Coordination Services having SUA Regions with “Status = Active”.

Steps 18, 19- FAA announces Radar Site Data Services for Radar Sites in SUA Region 1.

Step 20- FAA discovers Radar Site Data Services within active SUA Range.

Steps 21, 22, 23- FAA requests invocation of Radar Site Data Services for an active SUA Region.

Steps 23a, 23b- FAA requests invocation of Radar Site Data Services for active SUA Region, access granted, data received & displayed.

Step 24, 24a- DoD announces “Region 1 Status = Deactivated” via SUA Coordination Service.

Steps 24b, 24c- FAA discovers SUA Coordination Services having “Status = Deactivated”, and can now utilize the airspace.

3.7.1.1.2 Detailed Demonstration Scenario Steps

The following list provides the detailed activities that take place in in the step numbers shown (in red) on the sequence diagram in Figure 20.

1. Configuration of Domain Security Policies (Done prior to demo)
    a. DoD Domain
        i. Users (entered into Kerberos directly)
        ii. Roles
        iii. User I&A Credentials
        iv. Protected Data Resources (Radar Site Track Data)
        v. Security Policy Rules (<Role> can <read> <Radar Site Track Data>)
    b. FAA Domain
        i. Users (entered into Kerberos directly)
        ii. Roles
        iii. User I&A Credentials
        iv. Security Policy Rules (<Role> can <read> <Radar Site Track Data>)
2. Creation of new Service Type in both FAA & DoD Domains (Done prior to demo)
    a. SUA Coordination Service Type Schema
    b. Radar Site Data Service Type Schema
3. Specification of Trusted External Domains for each Service Type to be Announced (Before demo)
    a. DoD Domain adds FAA Domain as Trusted Recipient of SUA Coordination Service

    b. FAA Domain adds DoD Domain as Trusted Recipient of SUA Coordination Service
    c. DoD Domain adds FAA Domain as Trusted Recipient of Radar Site Data Service
4. DoD Domain Instantiates an SUA Coordination Service for a New SUA Region
    a. Specify SUA Coordination Service Attributes for a new SUA Region ID 1
        i. Lat/Long coordinates of SUA Region (rectangle)
        ii. Set the "SUA Region Status" field to "Proposed" (vs. Active, or Deactivated).
        iii. Set the "SUA Region Clear" field to "No" (vs. Yes).
    b. Add new entry to Local Service Registry for SUA Coordination Service for SUA Region ID 1
    c. Repeat steps (a) and (b) for 5 more SUA Regions in different geographic locations
5. DoD Domain Securely "Announces" the New SUA Coordination Service to Trusted External Domains (FAA)
    a. FAA Domain SSM 1 Receives new SUA Coordination Service Announcement for SUA Region ID 1
6. FAA Domain SSM 1 decodes SUA Coordination Service Announcement for SUA Region ID 1
    a. FAA SSM 1 pushes SUA Coordination Service announcement entry for new SUA Region ID 1 into local Service Registry
7. FAA SUA Management Application queries Local Service Registry for available Service Types
    a. New SUA Coordination Service type is available
8. FAA SUA Management Application initiates an SUA Coordination Service Request to identify the SUA Coordination Service announcements that best fit FAA SUA Management Application's desired criteria. (i.e. -- FAA SUA Manager is responsible for all SUA activity within a specific corridor.)
    a. Enters desired attributes (coordinates, other attributes), and submits SUA Coordination Service request to local SSM 1.
9. Local SSM 1 performs a service matching operation to determine "best fit" of requested attributes vs. available SUA Coordination Service announcements. SSM 1 returns a list of matches, ranked in ascending order.
10. FAA SUA Management Application selects the result set of SUA Coordination Service Announcements (pertaining to their region of responsibility), and:
    a. Must divert current flight routes around the proposed SUA Region. (not in demo)
    b. Take action to divert all affected flight paths. (not in demo)
    c. Verifies that the proposed SUA Region is clear. (not in demo)

d. Creates a new SUA Coordination Service Announcement for the proposed SUA region (having the original Service Announcement ID and region attributes), with the "SUA Region Clear" attribute field set to "Yes".

e. Announces the SUA Coordination Service Announcement update to the local Service Registry.

11. Local SSM 1 generates a new encrypted SUA Coordination Service Announcement that it sends to DoD Domain SSM 2 (given that Domain 2 SSM is in the Trusted External Domain List for SUA Coordination Service Announcements authorized to leave Domain 1).

12. DoD SSM 2 decrypts the SUA Coordination Service Announcement update and pushes into Domain 2 local Service Registry.

13. DoD SUA Management Application queries the local Service Registry for SUA Coordination Service Announcements matching the desired attribute criteria (i.e. -- for SUA Region IDs having the "Region Status" field set to "Proposed", and having the Region Clear field set "Yes").

    a. Registry returns matching results for Region 1 (for SUA Region IDs having the "Region Status" field set to "Proposed", and having the Region Clear field set "Yes").

14. DoD SSM 2 confirms the "Region Clear = Yes" status for the proposed SUA region, and starts the process to initiate military training exercises in SUA Region 1.

    a. DoD SSM 2 changes the SUA Coordination Service Attribute for SUA Region 1 Status from "Proposed", to "Active" in a new SUA Coordination Service Announcement for SUA Region 1, and pushes the updated SUA Coordination Service Announcement into its local Service Registry.

15. In parallel, DoD SSM 2 sends an encrypted SUA Coordination Service Announcement for Region 1 to SSM 1 in FAA Domain 1.

16. SSM 1 decrypts the announcement and adds the updated SUA Coordination Service Announcement into its local Service Registry.

17. FAA SUA Management Application queries registry for SUA Coordination Service Announcement updates, and finds that SUA Region 1 now has "Status = Active".

18. Meanwhile in DoD Domain 2, Radar Site Data Service must also be announced for the radar sites providing coverage of the newly activated SUA Region 1. (DoD is given access control authority for Radar Site Data Services associated with the active military SUA Region.) The DoD SUA Management Application provides a tool to identify which Radar Sites provide coverage in SUA Region 1's physical geographic area. A Radar Site Data Service Announcement is generated by the DoD SUA Management Application for each individual Radar Site within the newly activated SUA Region 1, and pushed to the Local Service Registry.

19. In parallel, SSM 2 creates an encrypted Radar Site Data Service Announcement for each Radar Site Track Data ID in SUA Region 1, and pushes the announcement to FAA Domain 1's SSM.

a. Domain 1 SSM decrypts each Radar Site Data Service Announcement and pushes it into its local Service Registry.

20. SSM 1 Admin GUI is utilized to request available Radar Site Data Services for all radar sites that reside within the SUA Manager's geographic area of responsibility. SSM 1 queries its local service registry and returns a set of matching results that includes the Radar Site Data Service Announcements for each Radar Site Track Data ID in SUA Region 1.

21. FAA SUA Management Application selects all of the Radar Site Data Services in SUA Region 1, and submits the Invocation Requests for each Radar Site Data Service to its local Service Invocation API.

22. For each of the (4) individual Radar Site Track Data Service Invocation Requests in SUA Region 1, SI 1 determines that it must forward them to SI 2 in Domain 2.

23. For each of the (4) individual Radar Site Track Data Service Invocation Requests in SUA Region 1, SI 2 in Domain 2 verifies the valid identity of the requestor (Domain 1), determines that the requested resources (Radar Site Data Services 1 through 4 for SUA Region 1) are in the local security domain, and verifies that the Requestor (Domain 1) has "read" privilege for the requested Radar Site Data Services. SI 2 grants access to the Requestor and provides a token (via SI 1) for access to each of the granted Radar Site Track Data Services.

    a. FAA SUA Management Application utilizes each token to perform the granted read operation (via its access Agent in Domain 1, which is transparently linked to a corresponding access Agent in Domain 2 that can access the granted Radar Site Data) for each of the 4 Radar Site Data Services in SUA Region 1, which are all located in Domain 2.

    b. The Data from each Radar Site in SUA Region 1 is displayed in the Domain 1's radar data display application.

24. DoD Domain 2 has completed military exercises in SUA Region 1 and needs to de-activate the SUA. SSM 2 Admin GUI is utilized to create an updated SUA Coordination Service Announcement for SUA Region 1, and change the "Region Status" field from "Active", to "De-Activated". The new SUA Coordination Service Announcement is pushed to the local service registry, and announced to the FAA Domain's SSM 1.

    a. FAA SSM 1 pushes the SUA Coordination Service Announcement into its local registry.

    b. FAA SSM 1 Admin queries the local FAA registry for "De-Activated" SUA Regions they are responsible for.

    c. FAA SSM 1 determines that SUA Region 1 has been de-activated and can now utilize the airspace in that region.

4 Results and Discussion

The primary objective of this effort was to advance the capability to automate the announcement, discovery, request matching, and life cycle management of information services both within and across security domain boundaries.

The accomplishments that were completed and documented to meet that objective include the following:

- **Technology gap description:** A research effort was conducted to identify gaps in the ability of current security models to meet the stated objective of this effort. It was determined that current Trust Models place trust in the user after exchange of security credentials, to directly access the service registry in another domain, and then directly access the desired service provider. This was identified as a vulnerability that needed to be mitigated via the technical approach in this effort.
- **New Paradigm to secure information services:** The technology gap identified in the research effort was used to derive a new methodology to secure the exchange of information services between security domains. The primary assertion is that security of information service management is not just about protecting the confidentiality (encryption) of information services, or distributing trust via user security tokens, but rather it is about protecting each entire security domain by not divulging free information that can be used to directly exploit weaknesses in the underlying infrastructure. The following two key steps were used to drive the technical design of a new secure service management framework:
    - Decouple the management of information services from the execution (invocation) of information services.
    - Use full location transparency to protect both the service management and the service invocation subsystems, by means of embedding the core security design principles from the ground-up.
- **Recommended service management architecture:** A service management architecture was designed to meet the core security principles embodied in the new paradigm. The service management architecture implemented the following principles:
    - Cross-Domain trust is point-to-point.
    - Each domain has control over which external domains it shares services with.
    - A Cross-Domain Service Announcement is always securely “pushed” to trusted external domains.
    - Encryption methodology for each Announcement is unique to the recipient domain, such that only the intended recipient domain can decrypt a specific Announcement.
    - Local Registries contain all services available to the Domain, but a Requestor does not know the specific location of the Service Provider.

- **Research contributions:** During the design of the service management architecture, the following contributions were made to help advance the current state-of-the-art in secure information service management:
    - A new Trust Model for Secure Service Management was defined and implemented.
    - A Secure Service Announcement methodology was defined and implemented, which contained the following:
        - Location transparency between service providers and requestors.
        - Domain-based trust for cross-domain services.
        - A secure cross-domain service announcement algorithm, with a novel method for Domain-specific encryption of announcements such that only intended recipient Domains could decrypt the announcements intended for them.
    - A Secure Service Manager system was defined and implemented, which included:
        - A method for management of secure service announcements.
        - A Service Requestor API for secure Lookup and issuance of weighted requests for available information services.
        - A Service Request matching capability to identify a rank-ordered "best-fit" for a service request against all currently available services that are in the system that may fulfill the desired request attributes.
        - A method to monitor and manage the lifecycle of an information service.
    - A Secure Private Registry was defined and implemented, enabling each security Domain to provide its users with local knowledge of all available information services (both internal and external to the local Domain), without divulging the location information of those services.
    - An open API for Secure Service Invocation Requests was defined and implemented, which enables the decoupling of information service management from the invocation of services.

Research prototype: A proof-of-concept prototype was designed and implemented in software to validate the secure service management concept using an operationally-realistic scenario.

The chosen scenario demonstrated a collaborative Community Of Interest (COI) business process, where the FAA in one security Domain, and the DoD in another security Domain, have shared responsibility for the coordinated establishment, safe utilization, and de-activation of Military Special Use Airspace (SUA) Ranges.

The Concept Demonstrator successfully demonstrated a Service Manager that supports a collaborative workflow between the two (FAA & DoD) security Domains via an “SUA Coordination Service”, and a “Radar Site Data Service” for selective data sharing between Domains.

5 Concluding Remarks

5.1 Lessons Learned

During the course of this effort, it became evident that there are distinct benefits gained (from both a security and interoperability perspective) from decoupling Service Management from Service Invocation. Figure 21 illustrates this concept. The Service Manager significantly aids collaborative workflows, where information services can be defined that support "stateful coordination" among peer applications in a COI that spans more than one security domain, as evidenced by the "SUA Coordination Service" example. This reduces the burden on having to force this type of peer-to-peer application collaboration into the Service Invocation system, which really only has to be responsible for securely brokering information services that must access data, such as the Radar Site Data via the "Radar Site Data Service" shown in the example.

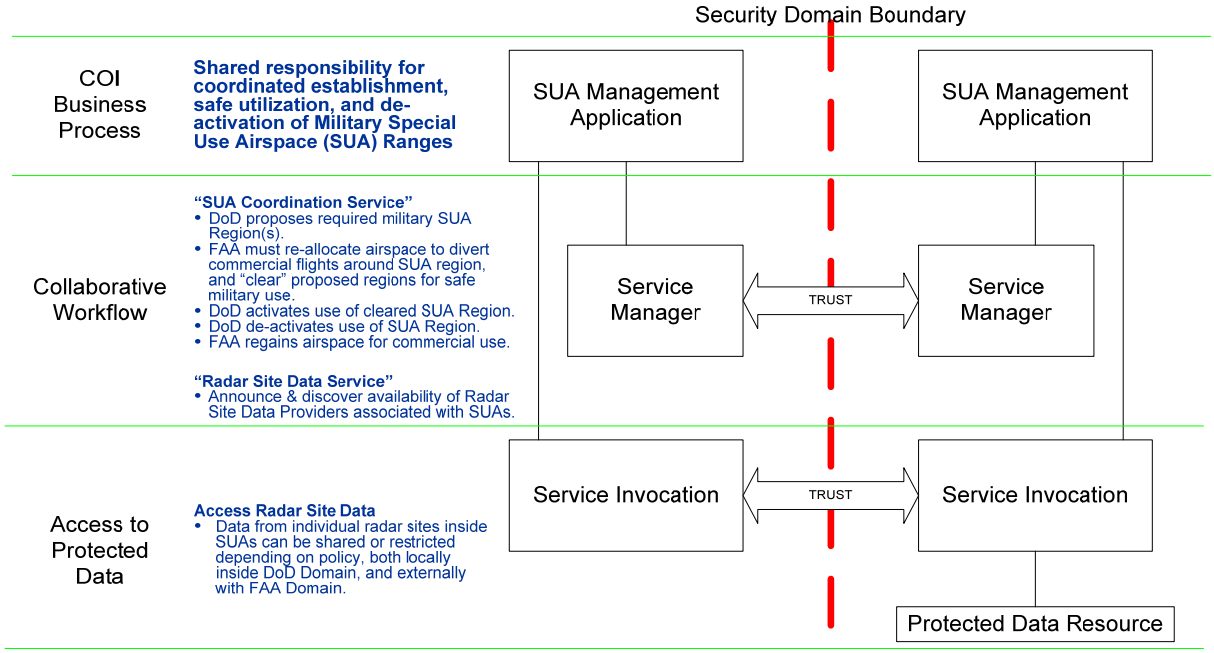


Figure 21: Decoupling Service Management and Service Invocation

Given that the development of a Service Invocation system is not part of the scope of this project, there is an important consideration that must be taken into account by the Government when utilizing their own GOTS implementation for a Service Invocation system in future integration efforts. To maintain the location transparency requirement for cross-domain security, the Service Manager (a trusted software process) must provide the Service Invocation system (another trusted software process) with the location information for the Domain to which a Service Invocation Request must be forwarded for fulfillment. This piece of information is outside of the API specified for the Service Invocation system in this effort, because it is intended to be implemented as a securely shared data item between these two trusted software components.

Recommendations

5.2 Future Work

5.2.1 Service Matching Algorithm

This effort identified a requirement for the Service Management system to provide a means to receive a Service Request (containing the desired values for each attribute within a Service Type that constitute what a Requestor "wants" to find in the Service Registry) and extract from the Service Registry the resulting "best-fit" in the form of a list of active services in the registry that most closely match the Request, ranked in order of closest fit.

The means for a requestor to identify their attribute preferences in terms of "weighted preference" for each attribute within a Service Type is independent of the Service Type. The service matching algorithm can rank the result set that it generates during the matching operation in a way that is also independent of the Service Type, but only if it considers the simple case where the total number of matching attributes is used as the single criterion for ranking the results. Future work can be done to provide a more robust service matching capability that provides the Requestor the ability to introduce more granular criteria for how they want to influence their result set, in a way that is more specific to the Service Type.

5.2.2 Automated Service Discovery

The method demonstrated in this effort for discovery of information services utilized a human-in-the-loop interface to more easily show a graphical approach for how a Requestor can interact with the Service Management system to view the contents of the Service Registry, and subsequently request information services. Future work can be done to automate the ability for an end user's software application to interact directly with the Service Management system to both request service, and respond to the result of the service matching operation, to invoke the desired service(s).

5.2.3 Advanced Technology Demonstration

This effort was focused on concept demonstration using a single use-case in a controlled laboratory environment. It is recommended that the next phase of development focus on integration of this technology with existing applications and information service providers, and demonstration in the context of current military use-cases to show the value-added to the warfigther.

6 Glossary of Terms

This Section contains a list of important terms used in this document, and their definitions.

Term	Definition
Domain	A (security) **Domain** in an SSM system is a collection of all Requestors, service providers, network devices, and physical network paths for which a Security Policy is defined and enforced. All data Targets in a Domain must be physically reachable via paths within the Domain. Different Domains have their own Security Policies running different instances of a Service Invocation System.
Information Service	An information service is an "offering" of structured data from a producer to eligible consumers. The service is described using a structured specification for what is "offered" by the producer, which we call a Service Specification.
Service Announcement	Service Announcement is concerned with how a service provider makes its information Service Specification known to the entities that are intended to access it.
Service Discovery	Service Discovery is concerned with how service consumers find the information services that they can potentially access and bind to. Beyond that, Service Discovery is more specifically concerned with finding the information services that match a set of Service Attributes that a consumer is interested in to satisfy their information needs.
Service Invocation	Service Invocation is concerned with actually instantiating a service between a provider and an authorized Requestor, subsequent to successful selection of Service Type and negotiation of Service Attributes.
Service Negotiation	Service Negotiation is a process through which a consumer's request for an information service is mapped to the specific provider or providers that can best satisfy all the attributes of the requested service. For a given Service Type, the service request processing system must map all service request attributes against the advertised attributes of active services, and determine a valid mapping that best fulfills the request. Service negotiation is concerned with the heuristics of how to make an appropriate mapping decision based upon each service type, and the attributes within that type that are negotiable during the mapping process.

Term	Definition
Service Release	Service release is concerned with termination of service according to terms of successful completion, or violation of, the negotiated attributes of the granted service.
Service Specification	A Service Specification is comprised of all necessary Attributes that define the characteristics of the Information Service offered. Service Specifications must be meaningful to the potential consumers of a service, such that consumers may unambiguously define their information needs in a request.

7 Symbols, Abbreviations, and Acronyms

Acronym/ Abbreviation	Definition
AFRL	Air Force Research Laboratory
API	Application Programming Interface
CI	Configuration Item
CM	Configuration Management
COI	Community of Interest
COTS	Commercial Off-The Shelf
CSCI	Computer Software Configuration Item
DoD	Department of Defense
GOTS	Government Off-The-Shelf
GUI	Graphical User Interface
HRS	Hardware Requirements Specification
HTTP	HyperText Transmission Protocol
HWCI	Hardware Configuration Item
ID	Identifier
IOR	Interoperable Object Reference
IP	Internet Protocol
JBI	Joint Battlespace Infosphere
NCES	Network-Centric Enterprise Services
PS	Product Specification

Acronym/ Abbreviation	Definition
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SRS	Software Requirements Specification
SSM	Secure Service Management
SVC	Service
TCP	Transmission Control Protocol
UDDI	Universal Description, Discovery & Integration
WS	Web Services
WSDL	Web Services Description Language
XML	eXtensible Markup Language